UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/934,407	08/21/2001	Cary Lee Bates	ROC920010085US1	8109

7590 05/02/2006

Gero G. McClellan
Thomason, Moser & Patterson, L.L.P.
Suite 1500
3040 Post Oak Boulevard
Houston, TX 77056-6582

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 05/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

e

Office Action Summary	**Application No.** 09/934,407	**Applicant(s)** BATES ET AL.	
	Examiner Michael J. Yigdall	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 02 February 2006.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-26 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-26 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is responsive to Applicant's submission filed on February 2, 2006. Claims 1-26 are pending.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

Applicant notes that in Olsen, the user selects a location at which to insert a breakpoint, and concludes that Olsen does not teach a method for a debugger application to select locations for inserting breakpoints in a program being debugged (remarks, page 8, first paragraph).

However, the user selecting a location at which to insert a breakpoint and the debugger selecting a location at which to insert a breakpoint are not mutually exclusive operations. One does not preclude the other. In Olsen, the user sets a breakpoint in the source code, and the debugger selects the correct location at which to insert the breakpoint in the machine code (see, for example, the abstract). In other words, Olsen teaches a method for the debugger to select a location for inserting a breakpoint in the program. The location that the debugger selects, in the machine code, corresponds to the location that the user selected in the source code. The plain language of the claims does not exclude these teachings of Olsen.

In response to Applicant's argument regarding breakpoints and tracepoints (remarks, pages 8-9), it is noted that breakpoints and tracepoints are not identical. However, breakpoints and tracepoints are indeed analogous. A breakpoint is inserted at a location within a program to trigger a debugging action. A tracepoint is inserted at a location within a program to trigger a debugging action. Shagam expressly discloses, "Software debugging approaches generally require manually inserting break or trace points at predetermined points within the source code" (column 1, lines 20-22, emphasis added). Furthermore, Shagam discloses, "The script generator creates a debugging script relatively quickly, even though a large number (e.g., 200) of trace points or break points are required within the source code" (column 1, lines 32-35, emphasis added). It is apparent that breakpoints and tracepoints are considered analogous in Shagam, even if each triggers a different debugging action.

Applicant contends that the program dependence graph of Gyimothy "fails to disclose anything regarding a block of execution at all, and is instead related to a graph of relationships among individual program statements" (remarks, page 10, top paragraph).

However, Olsen and Shagam define a "block" in the same manner as Applicant. For example, Olsen discloses, "A basic block of code is a sequence of code having one entrance point, one exit point and generally ends with a branch instruction" (column 5, lines 60-62). Likewise, Shagam discloses, "A block of execution is defined as a sequence of source level instructions with no flow control therein" (column 6, lines 25-26).

As set forth in the Office action mailed on November 2, 2005, Gyimothy teaches determining which other statements in a program affect or control execution of a statement (see, for example, page 4, first column, last paragraph). While this determination is performed at the statement level, it would have been obvious to one of ordinary skill in the art to apply these teachings to Olsen and Shagam at the block level, so as to determine which other blocks in the program affect or control execution of a block. As Applicant indicates, a block is a sequence of statements. The statements represented in the Gyimothy's program dependence graph form such blocks, and furthermore the suggestion is to apply the teachings of Gyimothy to the blocks disclosed in Olsen and/or Shagam.

One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,263,489 to Olsen et al. (art of record, "Olsen") in view of U.S. Patent No. 6,161,216 to Shagam (art of record, "Shagam") in view of *An Efficient Relevant Slicing Method for Debugging* by Gyimothy et al. (art of record, "Gyimothy").

With respect to claim 1 (currently amended), Olsen discloses a computer-implemented method for a debugger application (see, for example, debugger 30 in FIG. 1) to select locations for inserting breakpoints in a program being debugged (see, for example, the abstract, and steps 52 and 54 in FIG. 4A, which shows the debugger selecting locations for inserting breakpoints).

Although Olsen discloses selecting instructions in the program at which to insert breakpoints (see, for example, column 13, lines 8-12), Olsen does not expressly disclose:

(a) selecting branch points in the program being debugged at which to insert breakpoints.

However, Shagam discloses selecting control flow boundaries in a program at which to insert tracepoints (see, for example, column 2, lines 9-31). A branch is a control flow boundary. The tracepoints are analogous to breakpoints (see, for example, column 1, lines 20-22 and 32-25). Shagam discloses that the method for selecting the points in the program at which to insert the tracepoints or breakpoints improves the performance and accuracy of debugging (see, for example, column 1, lines 29-36 and 57-62).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the method of Olsen with the step of selecting branch points in the program at which to insert the breakpoints, so as improve the performance and accuracy of the debugging, such as taught by Shagam.

Olsen in view of Shagam further discloses the selecting comprising:

(i) identifying a statement for the program being debugged (see, for example, Olsen, column 12, lines 33-39, which shows identifying a statement);

(ii) determining a basic block that contains the statement (see, for example, Olsen, column 12, lines 28-32, which shows determining the basic block from which the statement is identified), wherein the basic block represents the sequence of consecutive program statements in which flow of control enters at the beginning and leaves at the end of the basic block (see, for example, Olsen, column 5, lines 60-62).

Although Olsen discloses analyzing the paths of execution that lead to the basic block based on control flow (see, for example, column 12, lines 20-24 and 40-45), Olsen in view of Shagam does not expressly disclose:

(iii) determining which other blocks present in the program control execution of the basic block.

However, Gyimothy discloses determining which other statements in a program affect or control execution of a statement (see, for example, page 4, first column, last paragraph), based on a program dependence graph that identifies control dependence among statements (see, for example, page 6, first column, Figure 5). Gyimothy discloses that the method for determining the relevant program slice ignores other statements that do not affect the statement and is accordingly applied to debugging (see, for example, page 1, first column, Abstract).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the method of Olsen in view of Shagam with the step of determining which other blocks in the program control execution of the basic block, so as to ignore blocks that do not control execution of the basic block, such as taught by Gyimothy.

Olsen in view of Shagam in view of Gyimothy further discloses:

(iv) inserting a breakpoint at each branch point contained in the other blocks present in the program that control execution of the basic block (see, for example, Shagam, column 4, lines 13-15, which shows inserting a tracepoint or breakpoint at each branch, and note that in view of Gyimothy, the other blocks that do not control execution of the basic block are ignored).

With respect to claim 2 (original), the rejection of claim 1 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein the statement is the location where the program being debugged halted execution (see, for example, Olsen, column 13, lines 13-20, which shows identifying the instruction or statement at which the program suspended or halted execution).

With respect to claim 3 (original), the rejection of claim 1 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein the blocks controlling execution of the basic block are blocks on which the basic block is control dependent (see, for example, Gyimothy, page 6, first column, Figure 5, which shows that the program dependence graph identifies control dependence).

With respect to claim 4 (previously presented), the rejection of claim 1 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein identifying the statement comprises identifying a statement in the source code of the program that may modify a program variable (see, for example, Gyimothy, page 1, first column, Introduction, which shows identifying statements that may influence or modify a variable).

With respect to claim 5 (original), the rejection of claim 4 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein identifying the statement that may modify a program variable comprises accessing a table comprising the variable mapped to statements that may modify the variable (see, for example, Gyimothy, page 5, first column, first paragraph, which shows a data set that maps a variable to the statements that affect the variable, and page 5, second column, which shows the data set in a table).

With respect to claim 6 (original), the rejection of claim 1 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein identifying the statement comprises identifying statements associated with loop latches (see, for example, Olsen, column 13, lines 48-60, which shows identifying statements associated with loops).

With respect to claim 7 (original), the rejection of claim 1 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein identifying statements associated with loop latches comprises accessing tables comprising the basic block mapped to the loop latches (see, for example, Olsen, column 8, lines 51-61, which shows a table for mapping source loops to basic blocks in the machine code).

With respect to claim 8 (original), the rejection of claim 1 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein identifying the statement comprises identifying a currently executing statement of each of a plurality of subprograms (see, for example, Olsen, column 7, lines 15-47, which shows identifying and executing statements of a plurality of functions or subprograms).

With respect to claim 9 (original), the rejection of claim 8 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein each of the plurality of subprograms is a portion of the program being debugged and performs a specific task (see, for example, Olsen, column 7, lines 15-47, which shows that the functions or subprograms in the program are debugged, and that the functions perform tasks that change the state of the program).

With respect to claim 10 (original), the rejection of claim 8 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein identifying the currently executing statement comprises accessing a table comprising the plurality of subprograms mapped to its respective currently executing statement (see, for example, Olsen, column 8, lines 43-50, which shows a table for mapping source constructs such the subprograms to machine code instructions that are executed).

With respect to claim 11 (previously presented), the rejection of claim 8 is incorporated, and Olsen in view of Shagam in view of Gyimothy further discloses the limitation wherein the blocks controlling execution of the basic block are blocks on which the basic block is control dependent (see, for example, Gyimothy, page 6, first column, Figure 5, which shows that the program dependence graph identifies control dependence).

With respect to claim 12 (currently amended), Olsen discloses a computer system comprising at least one processor configured to execute a debugging program (see, for example, CPU 12 and debugger 30 of computer system 10 in FIG. 1), wherein the processor, when executing the debugging program, is configured to perform an operation to select locations for inserting breakpoints in program being debugged (see, for example, the abstract, and steps 52 and 54 in FIG. 4A, which shows the debugger selecting locations for inserting breakpoints). The operation subsequently recited in the claim corresponds to the method of claim 1 (see the rejection of claim 1 above).

With respect to claim 13 (original), the limitations recited in the claim are analogous to those of claim 4 (see the rejection of claim 4 above).

With respect to claim 14 (original), the limitations recited in the claim are analogous to those of claim 6 (see the rejection of claim 6 above).

With respect to claim 15 (original), the limitations recited in the claim are analogous to those of claim 8 (see the rejection of claim 8 above).

With respect to claim 16 (currently amended), Olsen discloses a signal-bearing medium (see, for example, medium 36 in FIG. 1), comprising a program (see, for example, debugger 30 in FIG. 1) which, when executed by a processor (see, for example, CPU 12 in FIG. 1), performs an operation to select locations for inserting breakpoints in a program being debugged (see, for example, the abstract, and steps 52 and 54 in FIG. 4A, which shows the debugger selecting locations for inserting breakpoints). The operation subsequently recited in the claim corresponds to the method of claim 1 (see the rejection of claim 1 above).

With respect to claim 17 (original), the limitations recited in the claim are analogous to those of claim 2 (see the rejection of claim 2 above).

With respect to claim 18 (previously presented), the limitations recited in the claim are analogous to those of claim 3 (see the rejection of claim 3 above).

With respect to claim 19 (previously presented), the limitations recited in the claim are analogous to those of claim 4 (see the rejection of claim 4 above).

With respect to claim 20 (original), the limitations recited in the claim are analogous to those of claim 5 (see the rejection of claim 5 above).

With respect to claim 21 (original), the limitations recited in the claim are analogous to those of claim 6 (see the rejection of claim 6 above).

With respect to claim 22 (original), the limitations recited in the claim are analogous to those of claim 7 (see the rejection of claim 7 above).

With respect to claim 23 (original), the limitations recited in the claim are analogous to those of claim 8 (see the rejection of claim 8 above).

With respect to claim 24 (original), the limitations recited in the claim are analogous to those of claim 9 (see the rejection of claim 9 above).

With respect to claim 25 (original), the limitations recited in the claim are analogous to those of claim 10 (see the rejection of claim 10 above).

With respect to claim 26 (previously presented), the limitations recited in the claim are analogous to those of claim 11 (see the rejection of claim 11 above).

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure. See the attached Notice of References Cited.

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2192

mjy

[illegible]

TUAN DAM
SUPERVISORY PATENT EXAMINER